Defining SILF in K

$$
\begin{array}{rcl}
\mathit{Nat} & ::= & 0 \mid 1 \mid 2 \mid \ldots \text{ all natural numbers} \\
\mathit{Int} & ::= & \ldots \text{ all integer numbers} \\
\mathit{Bool} & ::= & \mathsf{true} \mid \mathsf{false} \\
\mathit{Val} & ::= & \mathit{Int} \mid \mathit{Bool} \\
\mathit{Name} & ::= & \text{all identifiers; to be used as names of variables and functions} \\
\mathit{Exp} & ::= & \mathit{Int} \mid \mathit{Bool} \mid \\
& \mid & \mathit{Name} \mid \mathit{Name}[\mathit{Exp}]\ [\mathit{strict}(2)] \\
& \mid & \mathsf{read}() \\
& \mid & \mathit{FunName}(\mathit{List}[\mathit{Exp}])\ [\mathit{ndstrict}] \\
& \mid & \mathit{Exp} + \mathit{Exp}\ [\mathit{ndstrict}\ \mathit{ext}\ \_ +_{\mathit{Int}} \_] \\
& \mid & \mathit{Exp} - \mathit{Exp}\ [\mathit{ndstrict}\ \mathit{ext}\ \_ -_{\mathit{Int}} \_] \\
& \mid & \mathit{Exp} * \mathit{Exp}\ [\mathit{ndstrict}\ \mathit{ext}\ \_ *_{\mathit{Int}} \_] \\
& \mid & \mathit{Exp}\,/\,\mathit{Exp}\ [\mathit{ndstrict}\ \mathit{ext}\ \_\mathit{quotient}_{\mathit{Int}}\_] \\
& \mid & \mathit{Exp}\,\%\,\mathit{Exp}\ [\mathit{ndstrict}\ \mathit{ext}\ \_\mathit{remainder}_{\mathit{Int}}\_] \\
& \mid & -\mathit{Exp}\ [\mathit{strict}\ \mathit{ext}\ -_{\mathit{Int}} \_] \\
& \mid & \mathit{Exp} < \mathit{Exp}\ [\mathit{ndstrict}\ \mathit{ext}\ \_ <_{\mathit{Int}} \_] \\
& \mid & \mathit{Exp} <= \mathit{Exp}\ [\mathit{ndstrict}\ \mathit{ext}\ \_ \leq_{\mathit{Int}} \_] \\
& \mid & \mathit{Exp} > \mathit{Exp}\ [\mathit{ndstrict}\ \mathit{ext}\ \_ >_{\mathit{Int}} \_] \\
& \mid & \mathit{Exp} >= \mathit{Exp}\ [\mathit{ndstrict}\ \mathit{ext}\ \_ \geq_{\mathit{Int}} \_] \\
& \mid & \mathit{Exp} == \mathit{Exp}\ [\mathit{ndstrict}\ \mathit{ext}\ \_ =_{\mathit{Int}} \_] \\
& \mid & \mathit{Exp}\ \mathsf{!=}\ \mathit{Exp}\ [\mathit{ndstrict}\ \mathit{ext}\ \_ \neq_{\mathit{Int}} \_] \\
& \mid & \mathit{Exp}\ \mathsf{and}\ \mathit{Exp}\ [\mathit{ndstrict}\ \mathit{ext}\ \_ \wedge_{\mathit{Bool}} \_] \\
& \mid & \mathit{Exp}\ \mathsf{or}\ \mathit{Exp}\ [\mathit{ndstrict}\ \mathit{ext}\ \_ \vee_{\mathit{Bool}} \_] \\
& \mid & \mathsf{not}\ \mathit{Exp}\ [\mathit{strict}\ \mathit{ext}\ \neg_{\mathit{Bool}}\_] \\
\mathit{Decl} & ::= & \mathit{Decl}; \mathit{Decl} \mid \mathsf{var}\ \mathit{Name} \mid \mathsf{var}\ \mathit{Name}[\mathit{Nat}] \\
\mathit{Stmt} & ::= & \mathit{Stmt}; \mathit{Stmt} \\
& \mid & \mathsf{write}(\mathit{Exp})\ [\mathit{strict}] \\
& \mid & \{\} \mid \{\mathit{Stmt}\} \mid \{\mathit{Decl}; \mathit{Stmt}\} \\
& \mid & \mathit{Name} = \mathit{Exp}\ [\mathit{strict}(2)] \\
& \mid & \mathit{Name}[\mathit{Exp}] = \mathit{Exp}\ [\mathit{ndstrict}(2,3)] \\
& \mid & \mathsf{if}\ \mathit{Exp}\ \mathsf{then}\ \mathit{Stmt} \mid \mathsf{if}\ \mathit{Exp}\ \mathsf{then}\ \mathit{Stmt}\ \mathsf{else}\ \mathit{Stmt}\ [\mathit{strict}(1)] \\
& \mid & \mathsf{while}\ \mathit{Exp}\ \mathsf{do}\ \mathit{Stmt} \mid \mathsf{for}\ \mathit{Name} = \mathit{Exp}\ \mathsf{to}\ \mathit{Exp}\ \mathsf{do}\ \mathit{Stmt} \\
& \mid & \mathsf{call}\ \mathit{Exp}\ [\mathit{strict}] \\
& \mid & \mathsf{return}\ \mathit{Exp}\ [\mathit{strict}] \\
\mathit{FunDecl} & ::= & \mathsf{function}\ \mathit{Name}(\mathit{List}[\mathit{Name}])\ \mathit{Stmt} \\
\mathit{Pgm} & ::= & \mathit{Set}[\mathit{FunDecl}] \mid \mathit{Decl}\ \mathit{Set}[\mathit{FunDecl}]\ [\mathit{strict}(1)]
\end{array}
$$

$$
\begin{array}{rcl}
\multicolumn{3}{l}{Val ::= \ldots \mid \lambda List[Name].Stmt} \\
\multicolumn{3}{l}{Exp, Decl, Stmt, Pgm \leq K} \\
\multicolumn{3}{l}{Val \leq KResult} \\
Env & ::= & Map[Name, Loc] \\
Store & ::= & Map[Loc, Val] \\
FunStack & ::= & List[Env \times K] \\
ConfigItem & ::= & k(K) \mid fstack(FunStack) \mid env(Env) \mid genv(Env) \\
& \mid & in(List[Int]) \mid out(List[Int]) \mid store(Store) \mid nextLoc(Loc) \\
Config & ::= & List[Int] \mid [\![K, List[Int]]\!] \mid [\![Set[ConfigItem]]\!]
\end{array}
$$

$$
[\![p, il]\!] = [\![k(p \curvearrowright run)\ fstack(\cdot)\ env(\cdot)\ genv(\cdot)\ in(il)\ out(\cdot)\ store(\cdot)\ nextLoc(loc(0))]\!]
$$

$$
k(\underbrace{\underbrace{run}_{\mathsf{call\ main()}}}_{\cdot})\ env(\underbrace{\rho}_{\cdot})\ genv(\underbrace{\cdot}_{\rho})
$$

$$
[\![\langle k(\cdot)\ out(il)\rangle]\!] = il
$$

$$
k(\underbrace{\mathsf{var}\ x\rangle}_{\cdot})\ env(\underbrace{\rho}_{\rho[x \leftarrow l]})\ nextLoc(\underbrace{l}_{l +_{Loc} 1})
$$

$$
k(\underbrace{\mathsf{var}\ x[n]\rangle}_{\cdot})\ env(\underbrace{\rho}_{\rho[x \leftarrow l]})\ nextLoc(\underbrace{l}_{l +_{Loc} n +_{Loc} 1})
$$

$$
k(\underbrace{\mathsf{function}\ f(xl)\ s\rangle}_{\cdot})\ env(\underbrace{\rho}_{\rho[f \leftarrow l]})\ store(\underbrace{\sigma}_{\sigma[l \leftarrow \lambda xl.s]})\ nextLoc(\underbrace{l}_{l +_{Loc} 1})
$$

$$
\mathsf{call}\ v = .
$$

$$
k(\underbrace{(\lambda xl.s)(vl) \curvearrowright k)}_{s})\ env(\underbrace{\rho}_{\rho'[xl \leftarrow l..(l +_{Loc} |xl|)]})\ genv(\rho')\ fstack(\underbrace{\cdot\rangle}_{(\rho, k)})\ nextLoc(\underbrace{l}_{l +_{Loc} |xl|})
$$

$$
k(\underbrace{\mathsf{return}\ v \curvearrowright \_)}_{v \curvearrowright k})\ fstack(\underbrace{(\rho, k)\rangle}_{\cdot})\ env(\underbrace{\_}_{\rho})
$$

$$
k(\underbrace{\underbrace{x}_{\sigma[\rho[x]]}\rangle}_{\cdot})\ env(\rho)\ store(\sigma)
$$

$$
k(\underbrace{\underbrace{x[n]}_{\sigma[\rho[x] +_{Loc} n]}\rangle}_{\cdot})\ env(\rho)\ store(\sigma)
$$

$$
k(\underbrace{\mathsf{read}()\rangle}_{i})\ in(\underbrace{i\rangle}_{\cdot})
$$

$$
k_1; k_2 = k_1 \curvearrowright k_2
$$

$$
k(\underbrace{\mathsf{write}\ i\rangle}_{\cdot})\ out\langle\underbrace{\cdot}_{i}\rangle
$$

$$
\{\} = .
$$

$$
\{s\} = s
$$

$$
k(\underbrace{\underbrace{\{d; s\}}_{d \curvearrowright s \curvearrowright restore(\rho)}\rangle}_{\cdot})\ env(\rho)
$$

$$
k(\underbrace{restore(\rho)\rangle}_{\cdot})\ env(\underbrace{\_}_{\rho})
$$

$$
k(\underbrace{x = v\rangle}_{\cdot})\ env(\rho)\ store(\underbrace{\sigma}_{\sigma[\rho[x] \leftarrow v]})
$$

$$
k(\underbrace{x[n] = v\rangle}_{\cdot})\ env(\rho)\ store(\underbrace{\sigma}_{\sigma[\rho[x] +_{Loc} n \leftarrow v]})
$$

$$
\mathsf{if}\ b\ \mathsf{then}\ s = \mathsf{if}\ b\ \mathsf{then}\ s\ \mathsf{else}\ \cdot
$$

$$
\mathsf{if\ true\ then}\ s_1\ \mathsf{else}\ s_2 = s_1
$$

$$
\mathsf{if\ false\ then}\ s_1\ \mathsf{else}\ s_2 = s_2
$$

$$
\mathsf{for}\ x = e_1\ \mathsf{to}\ e_2\ \mathsf{do}\ s = (x = e_1; \mathsf{while}\ x \leq e_2\ \mathsf{do}\ \{s; x = x + 1\})
$$

$$
k(\mathsf{while}\ b\ \mathsf{do}\ s\rangle = k(\mathsf{if}\ b\ \mathsf{then}\ (s; \mathsf{while}\ b\ \mathsf{do}\ s)\ \mathsf{else}\ \cdot\rangle
$$